

AMENDMENTS TO THE CLAIMS

Please replace all prior versions of the claims with the following claim listing:

Claims:

- 1-2. (Canceled)
3. (Currently Amended) The ~~active data type~~ variable of claim ~~[[2]]~~ 21, wherein processing by the first algorithm is suspended until the ~~particular~~ routine is finished running.
4. (Currently Amended) The ~~active data type~~ variable of claim 3, ~~further comprising: wherein the data type further comprises:~~
a second algorithm ~~associated with the active data type, the second algorithm being configured to be~~ that is automatically executed once the updated numerical value ~~associated with the active data type instance~~ has been set, the second algorithm processing the set value to generate value processing results.
5. (Currently Amended) The ~~active data type~~ variable of claim 4, wherein the setting of the value and the processing of the set value by the second algorithm is delayed until the ~~particular~~ routine returns, wherein once the ~~particular~~ routine returns, the second algorithm processes the set value to generate value processing results.
6. (Currently Amended) The ~~active data type~~ variable of claim ~~[[1]]~~ 21, wherein the ~~active~~ data type is a parameter being utilized by the computer program.
7. (Currently Amended) The ~~active data type~~ variable of claim ~~[[1]]~~ 21, wherein the ~~active~~ data type is a symbol being utilized by the computer program.
8. (Currently Amended) The ~~active data type~~ variable of claim ~~[[1]]~~ 21, wherein the ~~active~~ data type is a string format being utilized by the computer program, the string format specifying a format of a string data type instance associated with the string format, the string format including a format operation, the format operation specifying an operation associated with the string data type instance.

9-11. (Canceled)

12. (Currently Amended) The apparatus of claim [[11]] 22, wherein, once the first algorithm has determined the updated numerical value of the ~~active data type instance~~, processing by the first algorithm is suspended until the test routine is finished running.

13. (Currently Amended) The apparatus of claim 12, wherein the ~~active data type~~ further ~~has~~ comprises a second algorithm ~~associated therewith, the second algorithm~~ configured to be automatically executed once the updated numerical value ~~associated with the active data type instance~~ has been set, the second algorithm processing the set value to generate value processing results.

14. (Currently Amended) The apparatus of claim 13, wherein the setting of the value and the processing of the set value by the second algorithm is delayed until the ~~particular~~ test routine returns, and wherein once the ~~particular~~ test routine returns, the second algorithm processes the set value to generate value processing results.

15-17. (Canceled)

18. (Currently Amended) The method of claim [[17]] 23, ~~wherein the attempt to access the value of the active data type instance is made by a particular routine, and~~ wherein, once the first algorithm has ~~determined the value of the active data type instance~~ executed to process the numerical value for providing an updated numerical value, ~~processing by~~ suspending the first algorithm is suspended until the ~~particular~~ routine is finished running.

19. (Currently Amended) The method of claim 18, further comprising the ~~step of:~~

automatically executing a second algorithm ~~associated with the active data type~~ when [[a]] the updated numerical value ~~associated with the active data type~~

instance is set, the second algorithm processing the set value to generate value processing results.

20. (Currently Amended) The method of claim 19, further comprising: delaying the automatic executing of wherein processing by the second algorithm of the set value associated with the active data type is delayed until the particular routine returns.

21. (New) A variable for use in a computer program, the computer program configured to verify the integrity of an electrical circuit, the variable comprising:

- a data type defining the size, range, and type of numerical values to which the variable is limited;

- a name enabling the computer program to identify the variable; and

- a numerical value;

wherein the data type further comprises a first algorithm that is automatically executed when a routine of the computer program attempts to access the numerical value of the variable; and

wherein, before the routine accesses the numerical value, the first algorithm begins processing the numerical value for providing an updated numerical value of the variable.

22. (New) A computer program for verifying the integrity of an electrical circuit, the computer program accessing a variable, the variable comprising a data type defining the size, range, and type of numerical values to which the variable is limited, the variable further comprising a name, enabling the computer program to identify the variable, and a numerical value, the computer program comprising:

- logic configured to execute at least one test routine, wherein the data type further comprises a first algorithm that is automatically executed when the at least one test routine attempts to access the numerical value of the variable; and

- logic configured to execute the first algorithm before the at least one routine accesses the numerical value, the first algorithm processing the numerical value to provide an updated numerical value of the variable.

23. (New) A method for accessing a variable, the method comprising:

- providing a variable, the variable having a data type defining the size, range, and type of numerical values to which the variable is limited;
- naming the variable, wherein the variable name enables the computer program to identify the variable;
- providing the variable with a numerical value;
- incorporating a first algorithm into the data type, the first algorithm configured to automatically execute when a routine of the computer program attempts to access the numerical value of the variable;
- determining when the routine attempts to access the numerical value;
- executing the first algorithm to process the numerical value for providing an updated numerical value of the variable; and
- allowing the routine to access the updated numerical value.